AMENDMENTS TO THE SPECIFICATION

On page 4, please replace the paragraph starting at line 31 with the following paragraph:

In a dynamically typed programming environment, types are assigned to each data value in memory at runtime, rather than assigning a type to a static, syntactic entity in the program source code. The dynamically typed programming environment catches errors related to the misuse of values at the time the erroneous statement or expression is executed. In contrast, types are assigned to sets of values based on the program's source code in a statically typed programming environment. Static type disciplines operate on program source code rather than on the program execution. Therefore, in the statically typed programming environment, certain kinds of errors are detected without executing the program. The statically typed programming environment is provided, for example, in ~~Java~~ JAVA, from Sun Microsystems, Inc. of Palo Alto, California.

On page 9, please replace the paragraph starting at line 29 with the following paragraph:

The generated code is a textual or graphical description language. The generated code may be a high-level programming language such as M, C, C++, ~~Java~~ JAVA or Ada, or it may be low-level machine or assembly language. Furthermore, the code may be generated for use in creating Field Programmable Gate Arrays (FPGAs), Complex Programmable Logic Device (CPLD), or Application Specific Integrated Circuit (ASIC) device, in which case the generated code is often known as Hardware Description Language (HDL) including VHSIC Hardware Description Language (VHDL), Verilog, or other hardware description language.

On page 10, please replace the paragraph starting at line 17 with the following paragraph:

The filter object 110 processes equations representing the filter object 110 which include the state equation and output equation, as described above. FIGURE 3 is an exemplary block diagram representation illustrating in more detail the filter object 110 depicted in FIGURE 2. The filter object 110 includes a state equation processing unit 310 and an output equation processing unit 330. The state equation processing unit 310 receives input data U and the initial state stored in the memory 230. The state equation processing unit 310 processes the state equation and determines the state X of the filter object 110 in response to the initial state and the given input U of the filter object 110. The determined state of the filter is stored in the memory 230 and used as an initial state in processing next input data. The output equation processing unit ~~350~~ 330 receives input data U and the state of the filter stored in the memory 230. The output equation processing unit ~~350~~ 330 processes the output equation in response to the input U and the state X provided from the memory 230, and produces the output Y of the filter object 110.